

David Zhao Akeley

PhD Student at MIT CSAIL

| | | | |
|---------------|--|---------------|--------------------------------------|
| Primary Email | dza724[at]gmail.com | Undergraduate | UCLA (Samueli School of Engineering) |
| Work Email | akeley98[at]mit.edu | UCLA Degrees | BS Computer Science, BS Mathematics |
| SMS | 1-408-763-5241 | UCLA GPA | 3.762 |

Work History

MIT CSAIL – September 2024 - Present – PhD Student

New PhD student advised by [Jonathan Ragan-Kelley](#)

Nvidia Corporation – October 2020 - June 2024 – Developer Technology Engineer (IC3)

1. Collaborated with Nvidia Research on GPU algorithms for SDF-based¹ 2D computational geometry. “How to Accelerate 2D Shape Processing for Manufacturing and Planning” - [GTC 2023](#) (s51140)
2. Wrote Vulkan samples for `VK_EXT_graphics_pipeline_library`, `VK_KHR_timeline_semaphore`, `GL_KHR_shader_subgroup_shuffle`, `VK_NV_inherited_viewport_scissor`, and ray tracing extensions.
3. Designed `VK_NV_inherited_viewport_scissor` and implemented its driver and toolchain support.
4. Consulted with business partners on integrating Nvidia technology: DMM, `GL_NV_path_rendering`.

Sholari LLC – July - September 2019 – Contractor

1. Worked on a tumor growth and treatment educational tool written with the Unity game engine.
2. Implemented line graphs, waterfall (bar) plots, and the user interface for the timeline (graph x-axis).
3. Wrote a multithreaded C++11 plugin for visualizing tumors & immune system responses as particle clouds, and integrated it with the single-threaded C# Unity Engine.

Stanford University – June - September 2018 – Undergraduate Research Assistant

1. Helped prototype Aetherling, a Haskell-embedded domain-specific language for designing hardware image processing pipelines with automatic parallelization and static scheduling²; co-authored paper with [David Durst](#) (lead author), [Dr. Kayvon Fatahalian](#), and [Dr. Pat Hanrahan](#).
2. Implemented a functional simulator of an early prototype of Aetherling.
3. Revised the type system to remove impediments to parallelizing line buffers.³

<https://aetherling.org> (“Type-Directed Scheduling of Streaming Accelerators” - PLDI 2020)

MediocrePy – March - June 2017 – Independent Project

1. Created an optimized library for reducing stacks of telescope images to a single image using pixel means or medians and optional outlier rejection (sigma clipping) for noise reduction.
2. Designed and implemented a multithreaded C core with AVX vectorization, with an optional NumPy wrapper. Decreased runtime (compared to the Python code replaced) from hours to milliseconds.
3. Collaborated with [Dr. Zheng Cai](#), UC Santa Cruz Astrophysics.

<https://github.com/akeley98/MediocrePy>

Tsinghua University – July - August 2016 – Summer Intern

1. Designed a small library for fitting and plotting standard microlensing event light curves given discrete measurements of a star’s apparent magnitude (brightness) through time.
2. Used Python, C++, SciPy, and Matplotlib.
3. Collaborated with [Dr. Shude Mao](#), Tsinghua Department of Astronomy.

¹Signed Distance Field

²i.e. not using ready-valid hardware interfaces.

³A line buffer device reads in an image as a stream of pixel values and outputs rectangular “windows” of the image.

Jide Technology Co. – June - July 2015 – Summer Intern

1. Performed product testing for RemixOS, an Android derivative with a desktop-like interface.
2. Edited international marketing material and wrote user documentation in English.
3. Collaborated with Jason Zheng and Jeff Zhao (International Marketing Manager).

Other Projects

WebGL Jelly Cube Project

Simple mass-spring system simulation written with Javascript, WebAssembly, and WebGL 2.0 (for refractive and reflective effects). Earned third place in the UCLA computer graphics class contest, Fall 2017.⁴

<https://github.com/akeley98/JellyMcJelloFace>

<https://youtu.be/YwvMSeB6NzU>

Myricube – Vulkan Voxel Renderer

This proof-of-concept hybrid raycast/mesh voxel rendering was inspired by my discontent with Minecraft's low render distance and slow chunk updates, even on high settings.

Conventional voxel renderers such as Minecraft's work by converting voxel models into a mesh of triangles to draw, possibly with some optimizations like hidden face removal or merging coplanar triangles. This uses the GPU for its designed purpose, but as render distance increases, the GPU is bottlenecked by the large number of small triangles generated. Myricube uses conventional mesh rendering⁵ for nearby voxels but draws more distant voxels with an alternative raycasting renderer. This renderer subdivides the voxel model into cubic chunks, computes a minimal AABB⁶ for each chunk's visible voxels, and draws the AABBs themselves with a raycasting fragment shader that only checks for intersections within the drawn AABB. Dividing the model into AABBs reduces the renderer's memory bandwidth requirements and time wasted raycasting through empty space, without requiring any acceleration structures be built.

Additionally, the project leverages memory-mapped files and Vulkan asynchronous transfers to allow for low-latency, high-throughput model upload and real-time animation.

<https://github.com/akeley98/myricube>

Proposed gem5 Hardware Simulator Partial Bypassing Patch

This patch is refactored C++11 code that I wrote for my Advanced Computer Architecture course project.

In an out-of-order CPU, hardware bypassing allows instructions stalled waiting for a register to start execution as soon as the functional unit (ALU, memory port, etc.) writing to said register completes, without waiting for the actual commitment to register file. This is crucial for performance, so mainline gem5 simulates complete bypassing between all functional units in the simulated CPU. However, this is not practical to realize in modern superscalar hardware, as bypassing costs grow quadratically with the number of functional units. My patch splits the CPU's functional units into separate pools, with bypassing simulated only within pools. The commit-to-register-file delay is imposed for values communicated between pools.

The patch received positive code reviews; however, it doesn't look like there's enough interest in the feature to actually merge it into mainline gem5.

<https://gem5-review.googlesource.com/c/public/gem5/+27767>

<https://github.com/akeley98/FU-pools/blob/b4d291429edb6aa4988888656b6867ff99591b90/fu-pools.pdf>

⁴<https://www.facebook.com/vasilescu.alex/posts/10155206917936588>

⁵More precisely, instanced rendering to convert subsets of a list of voxels into a mesh to draw.

⁶Axis aligned bounding box – requires only six front-facing triangles to draw.

Appendix: UCLA Education – September 2017 - August 2020

| | First Major | Computer Science | Second Major | Mathematics | GPA | 3.762 (August 2020) |
|------|-------------|------------------|---|---------------|--|---------------------|
| | | Title | | Content Notes | | |
| 2017 | A+ | CS 174A | Intro to Computer Graphics | | See WebGL Jelly Cube Project | |
| | A+ | EE M16 | Digital Systems | | Verilog Lab | |
| | A+ | Math 115A | Linear Algebra | | | |
| | A | Math 170A | Probability Theory | | | |
| 2018 | A | CS 35L | Software Construction Lab | | POSIX basics (e.g. pthreads, bash) | |
| | A | CS M146 | Machine Learning | | | |
| | A | Math 110A | Algebra | | Ring Theory | |
| | A | Math 131AH | Analysis Honors | | Metric Spaces | |
| | A+ | CS 180 | Algorithms & Complexity | | | |
| | A | Engr 185EW | Art of Engineering Endeavors | | Writing Intensive Team Project | |
| | A | Math 131BH | Analysis Honors | | Derivation, Riemann Integration | |
| | B- | CS 131 | Programming Languages | | | |
| | B+ | CS M152A | Digital Design Lab | | Verilog Team Project | |
| | B+ | Math 110AH | Algebra Honors | | Group Theory | |
| 2019 | C | Math 120A | Differential Geometry | | | |
| | A- | CS 111 | Operating Systems Principles | | Focus on POSIX | |
| | A+ | CS 181 | Formal Languages & Automata | | Regex, CFG, Turing Machines, Decidability | |
| | A- | Math 132H | Complex Analysis Honors | | | |
| | A- | CS 161 | Fundamentals of Artificial Intelligence | | | |
| | A+ | EE M116C | Computer Systems Architecture | | | |
| | B+ | Math 111 | Theory of Numbers | | Overview of p-adic Numbers | |
| | A- | CS 118 | Computer Network Fundamentals | | | |
| | B | CS 130 | Software Engineering | | Java Team Project | |
| | A | Math 134 | Systems of Differential Equations | | | |
| 2020 | A+ | CS 133 | Parallel & Distributed Computing | | OpenMP, OpenCL, MPI, GPGPU, FPGA | |
| | A+ | CS 251A | Advanced Computer Architecture | | gem5 Hardware Sim Project, Graduate Course | |
| | A | Math 110BH | Algebra Honors | | Ring Theory, Module Theory | |
| | A | Math 110C | Algebra* | | Field Theory, Galois Theory | |
| | A | Math 115B | Linear Algebra | | | |
| | A | Math 142 | Mathematical Modeling | | | |
| | A+ | Math 167 | Game Theory | | | |

*There is no honors equivalent to the Galois Theory Course.

West Valley College Education – 2015-2017

GPA 4.0

Select Courses

| | Title | Content Notes |
|---------|-------------------------------|---------------------------|
| Math 4B | Differential Equations | |
| Math 19 | Discrete Mathematics | |
| Psych 2 | Experimental Psychophysiology | Experiment Design & Paper |
| Phys 4D | Modern Physics | Relativity |

Typeset in L^AT_EX

Fonts: Computer Modern, FreeSans (GNU FreeFont), Ubuntu Mono (Dalton Maag & Canonical Ltd.)